AMENDMENTS

In the Claims

1. (Currently amended) A computer implemented method comprising installing a driver package onto a computer using a side by side installer by:

receiving a driver package;

generating a strong name for the driver package; and

assigning a storage location for driver files associated with the driver package based on the strong name wherein the driver files include a driver image file; and

loading the driver image file from the storage location into a memory on the computer to link the computer to the peripheral device.

2. (Canceled)

3. (Original) A method according to Claim 1, wherein the driver files include at least one of a driver image file and a service name.

4. (Original) A method according to Claim 1, wherein generating the strong name for the driver package includes hashing at least one of the driver files associated with the driver package.

5. (Original) A method according to Claim 4, wherein the at least one driver file includes any one of a driver catalog file, a setup information file, and a driver image file.

6. (Original) A method according to Claim 4, wherein generating the strong name for the

driver package includes concatenating the hash of the at least one driver file to at least one of a driver catalog file and a public key from the driver package.

7. (Original) A method according to Claim 1, wherein assigning a storage location to driver files from the driver package in accordance with the strong name includes assigning at least one driver file associated with the driver package to a subdirectory in a driver store.

8. (Original) A method according to Claim 1, wherein assigning a storage location to data from the driver package in accordance with the strong name includes assigning service name parameters to a service name database.

9. (Currently amended) A computer implemented method, comprising:

generating a unique identity for respective driver packages to which commonly named driver files correspond on a computer; ~~and~~

assigning driver files associated with the individual driver packages to subdirectories in a common storage based on the unique identity wherein the driver files include a driver image file; and

loading the driver from one of the subdirectories into a memory on the computer to link the computer to the peripheral device.

10. (Canceled)

11. (Currently amended) A method according to Claim 9, wherein the driver files further include ~~at least one of a driver image file and~~ a service name.

12. (Original) A method according to Claim 9, wherein generating the unique identity for the respective driver packages to which commonly named driver files correspond includes hashing at least one driver file associated with the driver package.

13. (Original) A method according to Claim 12, wherein generating the unique identity for the respective driver packages to which commonly named driver files correspond includes appending a vendor's digital signature to the hash.

14. (Original) A method according to Claim 9, wherein assigning driver files from the respective driver packages to subdirectories in a common storage avoids overwriting previous versions of commonly named driver files.

15. (Original) A method according to Claim 11, wherein the assigning includes assigning driver image files from the respective driver packages to subdirectories in a driver store based on the unique identity for the driver package to which the respective driver image files correspond.

16. (Original) A method according to Claim 11, wherein the assigning includes assigning service names from the respective driver packages to a service name database that includes a service key and an image path corresponding to one of the driver files.

17. (Currently amended) A computer-readable storage medium having one or more instructions to be executed by one or more processors, the one or more instructions causing the one or more processors to:

generate a strong name for a driver package;

assign a storage location for a driver file associated with the driver package based on the strong name wherein the driver file is a driver image file; and

cause the one or more processors to further load the driver from the storage location into a memory to link the computer to the peripheral device.

18. (Canceled)

19. (Currently amended) A computer-readable storage medium according to Claim 17, wherein the strong name is a hash of at least one driver file associated with the driver package.

20. (Currently amended) A computer-readable storage medium according to Claim 17, wherein the strong name incorporates at least one of a driver catalog file and a public key corresponding to a vendor of the driver package.

21. (Currently amended) A computer-readable storage medium according to Claim 17, wherein the driver file is at least one of a driver image file and a co-installer, and wherein further the storage location is a driver store subdirectory.

22. (Currently amended) A computer-readable storage medium according to Claim 17, wherein the driver file is a service name, and wherein further the storage location is database to store a corresponding service key and image path.

23. (Currently amended) A computer-readable storage medium according to Claim 17, wherein the one or more instructions causing the one or more processors to assign a storage location for the driver file associated with the driver package refrains from assigning previously

assigned storage locations.

24. (Currently amended) [[~~An~~]] A computer implemented apparatus, comprising:

an installer to generate a strong name for a driver package; ~~and~~

a storage to install files associated with the driver package based on the strong name wherein the files associated with the driver package include a driver; and

wherein the apparatus further includes a loader to load the driver into a memory.

25. (Canceled)

26. (Original) An apparatus according to Claim 24, wherein the installer is to generate the strong name as a hash function of at least one driver file related to the driver package.

27. (Original) An apparatus according to Claim 24, wherein the installer is to generate the strong name by incorporating at least one of a digital signature and a public key related to the driver package.

28. (Original) An apparatus according to Claim 24, wherein the storage includes a subdirectory corresponding to the strong name.

29. (Currently amended) An apparatus according to Claim 28, wherein the storage is a driver store, and wherein further the files include ~~at least one of a driver image file and~~ a co-installer.

30. (Original) An apparatus according to Claim 28, wherein the storage is a database, and wherein further the files include a service name having a corresponding service key and image

path.

31. (Currently Amended) [[~~An~~]] A computer implemented apparatus, comprising:

means for generating a strong name for a driver package; ~~and~~

means for storing files associated with the driver package based on the strong name wherein the files associated with the driver package include a driver image file; and

means for loading the driver image file into a memory.

32. (Canceled)

33. (Original) An apparatus according to Claim 31, wherein the means for generating hashes at least one driver file from the driver package.

34. (Original) An apparatus according to Claim 31, wherein means for generating incorporates a digital signature related to the driver package into the strong name.

35. (Original) An apparatus according to Claim 31, wherein the means for storing has a subdirectory corresponding to the strong name.

36. (Original) An apparatus according to Claim 34, wherein the means for storing has a subdirectory for storing at least one of a driver package, a driver image file, and a co-installer.

37. (Original) An apparatus according to Claim 34, wherein the means for storing stores a service name having a corresponding service key and image path.

38. (Original) An apparatus according to Claim 34, wherein the means for storing stores a pointer from an operating system to a driver file for a particular device.